

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1-3. *(canceled)*

4. *(previously presented)* The fact extraction tool set of claim 7, wherein the attributes include tokenization, text normalization, part of speech tags, sentence boundaries, parse trees, and semantic attribute tagging.

5. *(previously presented)* The fact extraction tool set of claim 7, wherein the means for annotating the text comprises a plurality of independent annotators, wherein each of the annotators has at least one specific annotation function, and wherein the fact extraction tool set further comprises user-implemented means for specifying which of the annotators to use and the order of their use.

6. *(canceled)*

7. *(currently amended)* A fact extraction tool set for extracting information from a document, implemented using a client-server hardware architecture, wherein the document includes text, comprising:

means executed by the client-server hardware architecture for breaking the text into tokens;

a plurality of independent means executed by the client-server hardware architecture for annotating the text with token attributes, constituent attributes, links, and tree-based attributes, using XML as a basis for representing the annotated text, wherein each of the means for annotating has at least one specific annotation function; and

means executed by the client-server hardware architecture for resolving conflicting annotation boundaries in the annotated text to produce well-formed XML; and

means executed by the client-server hardware architecture for extracting facts from the annotated text using text pattern recognition rules, wherein each text pattern recognition rule comprises a pattern that describes text of interest, a label that names the pattern for testing and debugging purposes, and an action that indicates what should be done in response to a matching of the pattern, and wherein the text pattern recognition rules use regular expression-based functionality, tree-based traversal functionality based on a language that can navigate XML representations of text, and user-defined matching functions.

8. *(previously presented)* The fact extraction tool set of claim 7, wherein:

the token attributes have a one-per-base-token alignment, where for the attribute type represented, there is an attempt to assign an attribute to each base token;

the constituent attributes are assigned yes-no values, where the entire pattern of each base token is considered to be a single constituent with respect to some annotation value;
and

the links assign common identifiers to coreferring and other related patterns of base tokens.

9. *(canceled)*

10. *(canceled)*

11. *(previously presented)* The fact extraction tool set of claim 12, wherein the means for identifying and extracting potentially interesting pieces of information performs the further function of recognizing both true left and right constituent attributes and non-contiguous constituent attributes.

12. *(currently amended)* A fact extraction tool set for extracting information from a document, implemented using a client-server hardware architecture, wherein the document includes text, comprising:

means executed by the client-server hardware architecture for breaking the text into tokens;

a plurality of independent means executed by the client-server hardware architecture for annotating the text with token attributes, constituent attributes, links, and tree-based attributes, using XML as a basis for representing the annotated text, wherein each of the means for annotating has at least one specific annotation function;

means executed by the client-server hardware architecture for associating all annotations assigned to a particular piece of text with the base tokens for that text to generate aligned annotations; and

means executed by the client-server hardware architecture for identifying and extracting potentially interesting pieces of information in the aligned annotations by finding patterns in the attributes of the annotated text using text pattern recognition rules written in a rule-based information extraction language, wherein each text pattern recognition rule comprises a pattern that describes text of interest, a label that names the pattern for testing and debugging purposes, and an action that indicates what should be done in response to a matching of the pattern, and wherein the text pattern recognition rules use regular expression-based functionality, tree-based traversal functionality based on a language that can navigate XML representations of text, and user-defined matching functions, and wherein each text pattern recognition rule queries for at least one of literal text, attributes, and relationships found in the aligned annotations to define the facts to be extracted.

13-14. *(canceled)*

15. *(previously presented)* The fact extraction tool set of claim 12, wherein the user-defined matching functions are used to name and define a fragment of a pattern.

16. *(currently amended)* A machine readable storage having stored thereon a computer product application including a rule-based information extraction language for use in identifying and extracting potentially interesting pieces of information in aligned annotations in a text, the language comprising a plurality of text pattern recognition rules that query for at least one of literal text, attributes, and relationships found in the aligned annotations to define facts to be extracted, wherein each of the text pattern recognition rule comprises:

a pattern that describes text of interest;

a label that names the pattern for testing and debugging purposes; and

an action that indicates what should be done in response to a matching of the pattern; and

wherein the text pattern recognition rules use regular expression-based functionality, tree-based traversal functionality based on a language that can navigate XML representations of text, and user-defined matching functions.

17-18. *(canceled)*

19. *(previously presented)* The language of claim 16, wherein the user-defined matching functions are used to name and define a fragment of a pattern.

20. *(currently amended)* A text annotation tool implemented using a client-server hardware architecture, comprising:

means executed by the client-server hardware architecture for breaking text into its base tokens;

a plurality of independent annotators executed by the client-server hardware architecture for annotating the text with token attributes, constituent attributes, links, and tree-based attributes, using XML as a basis for representing the annotated text, wherein each of the annotators has at least one specific annotation function;

~~user implemented~~ means executed by the client-server hardware architecture for ~~specifying enabling a user to specify~~ which of the annotators to use and the order of their use;

means executed by the client-server hardware architecture for associating all annotations assigned to a particular piece of text with the base tokens for that particular piece of text to generate aligned annotations; and

means executed by the client-server hardware architecture for resolving conflicting annotation boundaries in the annotated text to produce well-formed XML.

21. *(previously presented)* The text annotation tool of claim 20, wherein the attributes include tokenization, text normalization, part of speech tags, sentence boundaries, parse trees, and semantic attribute tagging.

22-24. *(canceled)*

25. *(previously presented)* The text annotation tool of claim 20, wherein:

the token attributes have a one-per-base-token alignment, where for the attribute type represented, there is an attempt to assign an attribute to each base token;

the constituent attributes are assigned yes-no values, where the entire pattern of each base token is considered to be a single constituent with respect to some annotation value;
and

where the links assign common identifiers to coreferring and other related patterns of base tokens.

26-40. *(canceled)*

41. *(currently amended)* A method of extracting information from a document, implemented using a client-server hardware architecture, wherein the document includes text, comprising the steps of:

breaking the text into base tokens, using the client-server hardware architecture;

annotating the text with token attributes, constituent attributes, links, and tree-based attributes, using XML as a basis for representing the annotated text, using a plurality of independent annotators executed by the client-server hardware architecture, each of the annotators having at least one specific annotation function;

resolving conflicting annotation boundaries in the annotated text to produce well-formed XML, using the client-server hardware architecture; and

extracting facts from the annotated text using text pattern recognition rules written in rule-based information extraction language, using the client-server hardware architecture, wherein each text pattern recognition rule comprises a pattern that describes text of interest, a label that names the pattern for testing and debugging purposes, and an action that indicates what should be done in response to a matching of the pattern, and wherein the text pattern recognition rules use regular expression-based functionality, ~~tree-based~~ traversal functionality based on a language that can navigate XML representations of text, and user-defined matching functions. ~~tree-based~~ traversal functionality.

42. *(canceled)*

43. *(currently amended)* The method of claim 41, wherein ~~ain~~in the annotating step, the attributes include orthographic, syntactic, semantic, pragmatic and dictionary-based attributes.

44. *(canceled)*

45. *(currently amended)* The method of claim 41, wherein the annotating step is carried out by a plurality of independent annotators executed by the client-server hardware architecture, wherein each of the annotators has at least one specific annotation function, and wherein the method further comprises the step of allowing a user to specify which of the annotators to use and the order of their use.

46-47. *(canceled)*

48. *(previously presented)* The method of claim 41, wherein:

the token attributes have a one-per-base-token alignment, where for the attribute type represented, there is an attempt to assign an attribute to each base token;

the constituent attributes are assigned yes-no values, where the entire pattern of each base token is considered to be a single constituent with respect to some annotation value;
and

the links assign common identifiers to coreferring and other related patterns of base tokens.

49. *(currently amended)* The method of claim 41, wherein the annotating step includes associating all annotations assigned to a particular piece of text with the base tokens for that text to generate aligned annotations, using the client-server hardware architecture.

50. *(canceled)*

51. *(previously presented)* The method of claim 41, wherein the text pattern recognition rules have the ability to recognize both true left and right constituent attributes and non-contiguous constituent attributes.

52. *(previously presented)* The method of claim 41, wherein the text pattern recognition rules query for at least one of literal text, attributes, and relationships found in the aligned annotations to define the facts to be extracted.

53-55. *(canceled)*

56. *(previously presented)* The fact extraction tool set of claim 7, wherein the pattern recognition rules query for at least one of literal text, attributes, and relationships found in the annotated text to define facts to be extracted.

57. *(previously presented)* The fact extraction tool set of claim 7, wherein the means for annotating the text represents the annotated text as a single view of the document expressed as inline XML.

58. *(previously presented)* The fact extraction tool set of claim 12, wherein the means for annotating the text represents the annotated text as a single view of the document expressed as inline XML.

59. *(previously presented)* The method of claim 41, wherein in the annotating step, the annotated text is represented as a single view of the document expressed as inline XML.

60. *(previously presented)* The fact extraction tool set of claim 7, wherein the means for extracting uses XPath for traversing XML-based tree representations in the annotated text.

61. *(previously presented)* The fact extraction tool set of claim 12, wherein the means for extracting uses XPath for traversing XML-based tree representations in the annotated text.

62. *(previously presented)* The method of claim 41, wherein in the extracting step, XPath is used for traversing XML-based tree representations in the annotated text.